U.S. DEPARTMENT OF COMMERCE
ENVIRONMENTAL SCIENCE SERVICES ADMINISTRATION
WEATHER BUREAU
NATIONAL METEOROLOGICAL CENTER

DECEMBER 1969

REORGANIZATION OF FORECAST COMPUTING TO SUIT VECTOR MACHINES

OFFICE NOTE 34

Harold A. Bedient

DATA AUTOMATION DIVISION

Reorganization of Forecast Computing to Suit Vector Machines

The ability to compute large hydrodynamic problems in a reasonable time and at a reasonable cost is one of the limiting factors in the design of numerical models for daily operational use. This paper continues an earlier study of the restructuring of computing schemes to take advantage of new systems of computer organization. Reference throughout is made to a study made in conjunction with IBM of the RPQ 2938 Model 2 "Array Processor". This is a very fast "pipe-line" type computing box which simulates a much larger computer by doing a simple sequence of operations on a string of numbers and acts to the computer like an I-O device which can present sets of results. See Alsop[1]. In the subject study and for the sample problem, there were three kernels used for timing purposes. They were

Kernel 1: $A_i = B_i \pm C_i$

or

$$A_i = B_i * C_i \qquad i = 1,50$$

Kernel 2: $A_i = A_i + B_i * C_i \qquad i = 1,50$

Kernel 3: $A_i = \sum_{j=0} D_j B_{i-j} \qquad i = 1,50$

The use of these kernels in a code to carry out a simple barotropic forecast and the actual code for the timings is shown in the appendix[1].

Publication of this note prompted a study by Mr. Francis Balint and Mr. Jerry Kennedy of the ESSA Computer Division. They showed that the nature of the CDC 6600 allowed the simulation of these kernels very rapidly and perhaps in even better running time. The CDC 6600 has an instruction stack which, if carefully used, allows 27 register-to-register instructions to be performed in a loop without memory reference. It has several parallel arithmetic devices whose use can be overlapped, 24 registers to handle scratch numbers and its memory is partitioned so several memory references may be made at once. Loop overhead is expensive but there is facility for storing two numbers at once so it is advantageous to do calculations for two neighboring points in one loop[2].

Normally, this kind of optimization is not attempted on large codes because of the programming difficulty. The small number of kernels required made it possible to use "Macro" assembler language to program each kernel with its required options in the most efficient way and then use this new Macro instruction set to effectively write the code in a special higher level language.

With the cooperation of the Computer Division, recoding the NMC 6-level PE model was done. In addition to the three kernels stated, two more occurred frequently enough to be added

$$\text{Kernel 4: } A_i = \text{if } B_i \text{ then } 0 \text{ else } A_i \qquad i = 1,50$$

$$\text{Kernel 5: } A_i = A_i + B_i/C_i \qquad i = 1,50$$

Kernel 4 was required to handle problems encountered in moist models such as if clouds occurred, then heating in a layer at that point is zero. All conditional calculations were carried out for both sides of the condition and then the unwanted parts were set to zero, using kernel 4 and the two parts added together.

Everything in the forecast calculation for computing values of one row in the next step is set up in one subroutine. This subroutine is entirely coded in Macro's. In addition to the five stated kernels, there are machine language codes to calculate

$$A_i = B_i{**}C \text{ where C is real} \qquad \text{for } i = 1,50$$

$$A_i = \ln C_i \qquad \text{for } i = 1,50$$

by special accelerated polynomial techniques based on some work by Mr. Clarence Beal.

Outside the main forecast subroutine, there is a calculation for adjustment of dry and moist stability which were left in Fortran. The forecast subroutine also exists in a completely parallel Fortran form for comparison checking. During the preparation of the code, Dr. John Stackpole worked modifying the moisture calculation physics. This was incorporated at the same time. The whole system was put in operation in early October 1969.

Recently, extensive time testing has been done to determine if the timing was validated. The estimated timing is shown in Table I. Counts of occurrence of various kernels are shown in Table II. The assumption made in the Alsop paper was that the 6-level PE was about ten times the Alsop model. In fact, by actual count, the present model is 18 times the Alsop model exclusive of the mathematical functions. Also, for convenience, the Alsop model was estimated on a 50 by 50 grid whereas the actual calculation is done on a 57 by 53 grid, which is about 20 percent larger.

Totalizing the inner part of the calculation without the radiation or saturation vapor pressure calculation gives an estimate of 27 seconds per forecast hour. Some verifying runs are shown in Table III. The calculated inner loop time of 27 sec. should be compared with 18*Alsop*1.2. This is verified by an actual run in which the subroutine was called 2016 times on the same data. The timing on the run was 27.2 seconds.

During this project, successive improvements of the compilers available on the CDC 6600 have given a big improvement in object code. This is shown by a figure of 47 seconds per forecast hour as the time achieved by the latest version of Fortran Extended on this same code.

Table III also shows timing with radiation and with calculation of the saturation vapor pressure. In each hour, one time step is done with radiation, one with saturation vapor pressure and one with only the inner kernel. This gives the weighted value of 32 seconds per hour for the current run. The total "no I-O" calculation measures 40 seconds per hour. The 8-second discrepancy is in the Fortran calculations in the statistics and in the convective adjustment procedure. I-O loses about 10 seconds per hour that could be recovered by very careful work on the disk handler. This will be taken care of automatically if Extended Core Storage (ECS) is used. Our present run by this calculation is about 40 minutes forecast, 17 minutes output and 7 minutes initializing. Output could be cut in half by use of ECS so if complete use of ECS were made, about 15 minutes could be taken off the entire 48-hour forecast run.

This form of acceleration could be used in other modern computers. Mention has already been made of the Array Processor concept. The CDC 7600 has a different collection of internal hardware, so another Macro design would be more optimal and probably exceed the six to one factors estimated for the 7600. The program design is particularly apt for the CDC "Star" and the ILLIAC IV which have direct parallel operations. The following discussion is based on a study of the ILLIAC IV.

The original concept of the ILLIAC IV was of a two-dimensional computer that would carry out in hardware Richardson's concept of a "forecast factory". Costs have cut the concept down to a large one-dimensional array oriented computer. The one-quadrant machine being built for the project at the University of Illinois has one central control "CU" which can execute a limited set of instructions for loop counting and collective control. The CU also issues a string of commands which are gang executed in parallel by 64 PU's. The PU's have their own associated memories. They contain an index register, a couple of manipulative registers, a facility for shifting data laterally from PU to PU called routing and some very fast arithmetic units. The clock time is 50 ns, the memory cycle is 300 ns. Other data can be gathered from the ILLIAC IV literature. A careful estimate was made by ILLIAC programmers of the running time by kernels for each PU. The result is shown in Table I. This gives a weighted speed ratio of 2.1 of an ILLIAC IV PU of 2.1 over a CDC 6600. Since 64 points can be calculated at one, this gives an expected speed gain of 134 to one. This together with the time lost in the present disk handling of 6/5 could give a speed up of 150:1 (not all due to the design of the ILLIAC IV). It would be convenient if the grid array was a multiple of 64 but that is not required as the remaining

PU's can be working on the next row in a so-called skewed array storage. In a machine like this, the model could be allowed to grow by more than an order of magnitude and still be run in less than 2 minutes per forecast day and well within present fiscal resources. Encoding the model for this computer from this code would be a straight forward transliteration and would not be too difficult.

TABLE I

Timing estimates by kernel for various machines in microseconds:

Machine	2938*	CDC 6600	ILLIAC IV#
Kernel 1	2.4	2	.8
Kernel 2	3.2	2	.8
Kernel 3	1.6	2	1.2
Kernel 4		2	.6
Kernel 5		4	3.5

*Time for the 2938 was based on a preliminary specification as used by Alsop. IBM later revised the timing specification upward.

#Time for the ILLIAC IV was based on a hand calculation for operation by one PU and are considered conservative by present estimates. Based on the Kernel population in Table II, this gives an ILLIAC IV PU an estimated factor of 2.1 over the CDC 6600.

TABLE II

Kernel counts in the NMC-PE Model:

	6 Level PE*	Alsop's Barotropic	10* Alsop
Kernel 1	366	10	100
Kernel 2	298	10	100
Kernel 3	160	13	130
Kernel 4	17		
Kernel 5	13		
B_i**C	8		
ln B_i	13		

TABLE III

Timing of parts of the calculation:

	Calculated	Measured
Inner forecast in Fortran		45.2
Inner forecast	27 sec/hr.	
Inner forecast + Radiation		39 sec/hr.
Inner forecast + Saturation VP		35 sec/hr.
Weighted rate		32 sec/hr.
Total No I-O		40 sec/hr.
Total W I-O		50 sec/hr.
10* Alsop	14 sec/hr.	
18* Alsop *1.2	30 sec/hr.	